

What is claimed is:

- SUBA 37
1. A parallel processor comprising a plurality of processing means which perform mutually parallel processing on the basis of instructions written in programs and are capable of communicating with each other via a common bus, wherein
- one of said processing means suspends processing based on a said program and enters a waiting state when executing a wait instruction ("sleep") and releases said waiting state and restarts the processing based on said program based on execution of a wait release instruction ("cont") by another processing means and
- the other processing means executes a next instruction without suspending processing after it executes said wait release instruction.
2. A parallel processor as set forth in claim 1, wherein said other processing means executes a synchronization wait instruction ("wait") to enter a synchronization waiting state and releases said synchronization waiting state based on execution of said wait instruction corresponding to said synchronization wait instruction or execution of a program end instruction ("end") indicating an end of a program by said one processing means.

Sub A4

3. A parallel processor as set forth in claim 1, further comprising:

a first storage means connected to said common bus and storing said programs and

5 second storage means provided corresponding to said plurality of processing means, reading from said first storage means programs to be executed by corresponding processing means via said common bus, supplying said processing means with instructions written
10 in the read programs, and having faster access speeds than said first storage means,

a said second storage means continuing to store a program supplied to its corresponding processing mean before entering said waiting state when said
15 processing means is in said waiting state.

4. A parallel processor as set forth in claim 3, wherein a said second storage means continues to store a program until its corresponding processing means executes a program end instruction indicating an end of a program.

20 5. A parallel processor as set forth in claim 3, wherein:

said other processing means executes a program execution instruction ("gen") to make said one processing means execute a program stored in said first
25 storage means; and

a said second storage means corresponding to said one processing means reads and stores a program which is instructed by said program execution instruction to be executed from said first storage means.

5 6. A parallel processor as set forth in claim 5, further comprising a program execution assigning means for determining said one processing means to execute a program instructed to be executed by said program execution instruction and for reading the program
10 instructed to be executed by said program execution instruction from said first storage means to said second storage means corresponding to said determined processing means.

15 7. A parallel processor as set forth in claim 5, wherein when said one processing means enters a waiting state based on said wait instruction, said other processing means which executed said program execution instruction executes said wait release instruction.

20 8. A parallel processor as set forth in claim 5, wherein when said one processing means enters a waiting state based on said wait instruction, a processing means other than said other processing means which executed said program execution instruction executes said waiting release instruction.

25 9. A parallel processor as set forth in claim 1,

wherein said plurality of processing means and a common bus for connecting the plurality of processing means are installed in a single semiconductor chip.

Sub A5
10. A parallel processor comprising a plurality
5 of processing means which perform mutually parallel processing on the basis of instructions written in programs and are capable of communicating with each other via a common bus, wherein

one of said processing means suspends
10 processing based on a said program and enters a waiting state when executing a wait instruction ("sleep") and releases said waiting state and restarts the processing based on said program based on execution of a wait release instruction ("cont_a") by another processing
15 means and

said other processing means enters a
synchronization waiting state when executing said wait
release instruction until said one processing means
enters said waiting state when said one processing means
20 is not in said waiting state.

11. A parallel processor as set forth in claim
10, wherein said other processing means executes a next
instruction without suspending its processing after
executing said wait release instruction when said one
25 processing means is in said waiting state when it

executes said wait release instruction.

Sub A6 12. A parallel processor as set forth in claim 10, further comprising:

5 a first storage means connected to said common bus for storing said programs;
second storage means provided corresponding to said plurality of processing means, reading from said first storage means programs to be executed by corresponding processing means via said common bus,
10 supplying said processing means with instructions written in the read programs, and having faster access speeds than said first storage means,

20 a said second storage means continuing to store a program supplied to its corresponding processing mean before entering said waiting state when said processing means is in said waiting state.

13. A parallel processor as set forth in claim 12, wherein a said second storage means continues to store a program until its corresponding processing means
20 executes a program end instruction ("end") indicating an end of a program.

14. A parallel processor comprising a plurality of processing means which perform mutually parallel processing on the basis of instructions written in
25 programs and are capable of communicating with each other

00120798-104885

via a common bus, comprising:

a first storage means connected to said common bus for storing said programs and

second storage means provided corresponding
5 to said plurality of processing means, reading from said first storage means programs to be executed by corresponding processing means via said common bus, supplying said processing means with instructions written in the read programs, and having faster access speeds
10 than said first storage means;

one of said processing means suspending processing based on a said program and entering a waiting state when executing a wait instruction ("sleep") and releasing said waiting state and restarting the
15 processing based on said program based on execution of a wait release instruction ("cont") by another processing means;

a said second storage means continuing to store a program supplied to its corresponding processing
20 mean before entering said waiting state when said one processing means is in said waiting state.

15. A parallel processor as set forth in claim 14, wherein a said second storage means continues to store a program until its corresponding processing means
25 executes a program end instruction ("end") indicating an

end of a program.

16. A parallel processor as set forth in claim 14, wherein:

5 said other processing means executes a program execution instruction ("gen") to make said one processing means execute a program stored in said first storage means; and

10 a said second storage means corresponding to said one processing means reads and stores the program instructed to be executed by said program execution instruction from said first storage means.

17. A parallel processor, as set forth in claim 16, further comprising a program execution assigning means for determining said one processing means to
15 execute a program instructed to be executed by said program execution instruction and for reading the program instructed to be executed by said program execution instruction from said first storage means to said second storage means corresponding to said determined processing
20 means.

Sub A1 18. A parallel processing method for performing at least first processing and second processing in parallel based on instructions written in programs, wherein:

25 said first processing suspends processing

based on a said program and enters a waiting state by
executing a wait instruction ("sleep") and releases said
waiting state and resumes processing based on said
program based on execution of a wait release instruction
5 ("cont") in said second processing and

said second processing executes a next
instruction without suspending its processing after
executing said wait release instruction.

19. A parallel processing method as set forth in
10 claim 18, wherein said second processing enters a
synchronization waiting state by execution of a
synchronization wait instruction ("wait") and releases
said synchronization waiting state based on execution of
said wait instruction corresponding to said
15 synchronization wait instruction in said first
processing.

Sub 18
20. A parallel processing method for performing
at least first processing and second processing in
parallel based on instructions written in programs,
20 wherein:

said first processing suspends processing
based on a said program and enters a waiting state by
executing a wait instruction ("sleep") and releases said
waiting state and resumes processing based on said
25 program based on execution of a wait release instruction

("cont_a") in the second processing and

said second processing enters a
synchronization waiting state by executing said wait
release instruction until said first processing enters
5 said waiting state when said first processing is not in
said waiting state.

21. A storage medium for storing in a computer-
readable format routines of first processing and second
processing to be performed in parallel based on
10 instructions written in programs, wherein:

said first processing is processing which
suspends processing based on a said program and enters a
waiting state by executing a wait instruction ("wait")
and releases said waiting state and resumes processing
15 based on said program based on execution of a wait
release instruction ("cont") in said second processing
and

said second processing is processing which
executes a next instruction without suspending its
20 processing after executing said wait release instruction.

22. A storage medium for storing in a computer-
readable format routines of first processing and second
processing to be performed in parallel based on
instructions written in programs, wherein:

25 said first processing is processing which

5 release instruction ("cont_a") in the second processing
and

10 said second processing is processing which
enters a synchronization waiting state by executing said
wait release instruction until said first processing
enters said waiting state when said first processing is
not in said waiting state.